

REMARKS

Claims 1, 3-8, 10-15 and 17-20 were pending at the time of examination. No claims have been amended or canceled. No new matter has been added. The Applicant respectfully requests reconsideration based on the foregoing amendments and these remarks.

Claim Rejections – 35 U.S.C. § 103

Claims 1-20 were rejected under 35 U.S.C § 103(a) as being unpatentable over U.S. Patent No. 5,724,570 to Zeller et al (hereinafter “Zeller”), in view of U.S Patent No. 6,480,833 to Kaneko et al (hereinafter “Kaneko”). The Applicant respectfully traverses these rejections.

The features and benefits of the Applicant’s invention, and the general differences compared to Zeller have been extensively discussed in previous Office Action responses and will thus not be repeated here. Instead, the following discussion will focus on the specific limitations of claim 1 and some of the dependent claims, and how the Applicant’s invention as claimed differs from the Zeller/Kaneko combination.

On page 4 of the Office Action, the Examiner contends that the limitation “storing the acquired datatype of the converted SQL template with the SQL template” is shown in Zeller col. 7, lines 1-17. The Applicant respectfully disagrees. What is shown in the cited section of Zeller is storing on a storage medium, such as a secondary storage memory hard drive F100, of Zeller’s “optimized tree.” No datatype is stored for the optimized tree. Instead the datatype is determined – supposedly in a faster way than for conventional trees due to the optimization - when the optimized tree is accessed by the executor F110. Thus, the “storing...” step of claim 1 is neither anticipated nor rendered obvious by the cited sections of Zeller.

Further, the Examiner agrees with the Applicant that the use of a cast function is not explicitly taught by Zeller, and relies on Kaneko for this showing. The Examiner contends on pages 4 and 5 of the Office Action and with some general references to figures 9 and 14, that Kaneko discloses

“overloading of casting function and determinations of datatypes of input variables as well as the entire SQL statement. ‘Overloading a function’ is a process by which on determination of the number and datatype of variables in calling a function, a particular function is selected to be executed. A determination of datatype must occur in order for the correct function to be selected...”

The Applicant respectfully disagrees with this statement. It is true that Kaneko discusses “overloading.” However, this overloading is not an overloading of casting functions, but simply referred to in Kaneko as an overloading of “routines.” In fact, the casting function is never

mentioned in Kaneko. It should also be noted that Kaneko nowhere mentions any determination of datatypes. Throughout Kaneko there are statements making it clear that the datatype is defined by the user (see, for example, col. 1, lines 23-24; “*...the data type is called the abstract data type which is defined by a user...*”).

By reading the Kaneko specification it is also clear that a determination of datatype is not necessary for the correct function to be selected, as the Examiner alleges. Instead the proper function is selected based on a “data type precedence list 108,” which defines data type inheritance hierarchy (see, for example, col. 2, lines 26-43), in combination with “routine invocation information,” which is described with respect to FIG. 5 and in the specification at col. 11, line 60 – col. 12, line 57. There is little point in going into the specific details of this procedure, but the main point here is to note that there is no data type determination in Kaneko. Instead, Kaneko, through the variable declarations by the users and the defined data type inheritance hierarchy, immediately knows which function to apply without doing any further determinations (or applying any casting functions).

Thus, it is clear that Kaneko does not anticipate or render obvious the use of a casting function, as recited in claim 1. If the Examiner disagrees, he is kindly requested to provide references to more specific portions of Kaneko that support his contrary allegations. For at least the reasons above, it is respectfully submitted that the Applicant’s invention, as defined in claim 1, is patentably distinct from the Zeller/Kaneko combination.

Claims 3-7 depend directly or indirectly from claim 1, and are thus not anticipated or rendered obvious for at least the reasons discussed above. Further, with respect to claims 3 and 6 in particular, the Examiner appears to be drawing his own conclusions and citing the same general section that was cited with respect to claim 1, that is, col. 7, lines 1-17. This section appears to neither show any specific support for the limitations “forming a valid SQL statement from the converted SQL template” in claim 3, nor for “inquiring if a descendent of the converted SQL template has been modified” of claim 6. It is respectfully requested that the Examiner provide more specific support in Zeller for these rejections, or a more detailed basis of his reasoning, as Zeller does not teach converted SQL templates, as defined in the claims. Thus, it is respectfully requested that the rejection under 35 U.S.C § 103(a) be withdrawn for dependent claims 3-7.

For reasons substantially similar to those set forth above, the Applicant respectfully contends that the rejection of the computer program product claims 8 and 10-14 and the system claims 15 and 17-20 is unsupported by the cited art and should be withdrawn.

Conclusion

The Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
MOLLBORN PATENTS, INC.

/Fredrik Mollborn/

Fredrik Mollborn
Reg. No. 48,587

2840 Colby Drive
Boulder, CO 80305
(303) 459-4527